

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-137684

(43)Date of publication of application : 31.05.1996

(51)Int.Cl. G06F 9/06  
G06F 9/06

(21)Application number : 06-278945 (71)Applicant : SHARP CORP

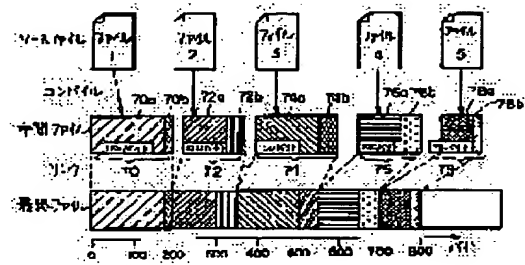
(22)Date of filing : 14.11.1994 (72)Inventor : TANAKA TOSHIYUKI

## (54) PROGRAM TRANSFER METHOD, SYSTEM AND DEVICE, FOR PROGRAM DEVELOPMENT, AND PROGRAM EXECUTION DEVICE

### (57)Abstract:

**PURPOSE:** To reduce the transfer time by transferring only an object file subject to revision to a target device again and linking the object file with other object file having been transferred in an order.

**CONSTITUTION:** Intermediate object programs being the result of compiling files 1 to 5 are intermediate files 70a, 72a, 74a, 76a and 78a. Then unused areas 70b to 78b are provided in advance to the object files 70a to 78a. When, e.g. the file 72a is corrected, only the object program after correction is transferred to the program execution device. Then the transferred object file is linked in the order that the object file has the same address in the execution form program as that of the address before the revision. Thus, the executable form program after the revision is generated in the program execution device by having only to link the revised and transferred object file with the other object files having been transferred in an order.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平8-137684

(43) 公開日 平成8年(1996)5月31日

(51) Int.Cl. <sup>5</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 9/06	5 4 0 A	7230-5B		
	S	7230-5B		
	4 1 0 E	7230-5B		

審査請求 未請求 請求項の数 8 O L (全 15 頁)

(21) 出願番号 特願平6-278945

(22) 出願日 平成6年(1994)11月14日

(71) 出願人 000005049

シャープ株式会社

大阪府大阪市阿倍野区長池町22番22号

(72) 発明者 田中 敏幸

大阪府大阪市阿倍野区長池町22番22号 シ

ャープ株式会社内

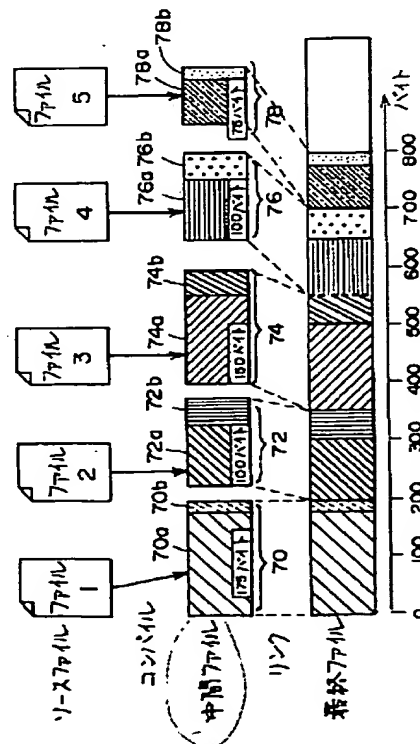
(74) 代理人 弁理士 深見 久郎

(54) 【発明の名称】 プログラム転送方法、プログラム開発システムならびにプログラム開発装置およびプログラム実行装置

(57) 【要約】

【目的】 プログラムの開発時間を削減する。

【構成】 プログラム開発装置からプログラム実行装置に転送する最終ファイルにおいて、予め各中間オブジェクトファイル70a~78aに未使用領域70b~78bを付加しておく。そしてたとえばファイル72aを修正した場合、修正後のオブジェクトプログラムのみをプログラム実行装置に転送する。修正後のオブジェクトプログラムの長さが修正前のオブジェクトファイル72より短ければ、修正後オブジェクトプログラムの最終部分を、予め付加されていた未使用領域72bまで延ばしてリンクする。修正後のオブジェクトプログラムの長さが変更前のオブジェクトファイル72よりも長ければ、変更後オブジェクトプログラムは、最終ファイルの最後にリンクする。



## 【特許請求の範囲】

【請求項 1】 相互にリンクされ、かつ各々がその内部に未使用領域を有する複数のオブジェクトファイルからなる実行可能形式のプログラムをプログラム開発装置からプログラム実行装置に転送し、前記プログラム実行装置の記憶装置上に展開する第 1 の転送ステップと、前記複数のオブジェクトファイルの少なくとも 1 つを変更して、変更前の大きさと予め定める関係を有する大きさの変更後オブジェクトファイルを前記プログラム開発装置において作成し、変更されたオブジェクトファイルを含む前記複数のオブジェクトファイルを、前記所定の順番と予め定める関係にある順番に従ってリンクして変更後の実行可能形式のプログラムを作成するステップとを含み、前記予め定める関係にある順番は、前記複数のオブジェクトファイルのうちの変更されないものが、前記変更後の実行可能形式のプログラム前において変更前と一致する位置となるように選択されており、さらに、前記変更後の実行可能形式プログラムのうち、変更されたオブジェクトファイルのみを前記プログラム実行装置に再び転送し、他の既に転送済みの変更されないオブジェクトファイルと前記所定の順番と予め定める関係にある前記順番に従ってリンクするように前記記憶装置上に展開する第 2 の転送ステップを含む、プログラム転送方法。

【請求項 2】 前記作成するステップは、前記変更されるオブジェクトファイルに対応するソースプログラムをコンパイルして、オブジェクトプログラムを作成し、かつ前記変更後オブジェクトファイルの大きさが、予め定める単位長の倍数となるように、前記オブジェクトプログラムの末尾に所定長以上の大きさの未使用領域を付加して前記変更後オブジェクトファイルを生成するステップと、変更されたオブジェクトファイルを含む前記複数のオブジェクトファイルを、前記所定の順番と予め定める関係にある前記順番に従ってリンクして変更後の実行可能形式のプログラムを作成するステップとを含み、前記予め定める関係にある前記順番は、前記変更後オブジェクトファイルの大きさが変更前の大きさ以下であるときは前記第 1 の転送ステップにおいて転送される実行可能形式プログラムにおける順番と一致するように選択され、前記変更後オブジェクトファイルの大きさが変更前の大きさよりも大きいときには、前記変更後オブジェクトファイルが他のすべてのオブジェクトファイルの末尾に位置するように選択される、請求項 1 記載のプログラム転送方法。

【請求項 3】 前記作成するステップは、前記変更されるオブジェクトファイルに対応するソースプログラムをコンパイルして、オブジェクトプログラムを作成するステップと、

変更されたオブジェクトプログラムを含む、前記複数のオブジェクトファイルに格納される複数のオブジェクトプログラムを、前記所定の順番と予め定める関係にある順番に従って、かつ前記複数のオブジェクトプログラムのうちの変更されないものが、前記変更後の実行可能形式のプログラム内において変更前と一致する位置となるように、隣接するオブジェクトプログラム間に未使用領域が設けられるようにリンクして、変更後の実行可能形式のプログラムを作成するステップとを含み、前記予め定める関係にある順番は、前記変更されたオブジェクトプログラムの大きさが変更前のオブジェクトファイルの大きさより小さいときは前記第 1 の転送ステップにおいて転送される実行可能形式プログラムにおける順番と一致するように選択され、前記変更されたオブジェクトプログラムの大きさが変更前のオブジェクトファイルの大きさ以上であるときには、前記変更後オブジェクトプログラムが他のすべてのオブジェクトファイルの末尾に位置するように選択される、請求項 1 記載のプログラム転送方法。

【請求項 4】 オブジェクトファイルに変更がなされたか否かを、当該オブジェクトファイルが前回転送されたときの日付と、当該オブジェクトファイルの作成日付とを比較することにより行なうことを特徴とする、請求項 1 から 3 のいずれかに記載のプログラム転送方法。

【請求項 5】 前記第 2 の転送ステップにおいて、変更されたオブジェクトファイルを変更前オブジェクトファイルと比較し、変更のあった部分のみを前記プログラム実行装置に転送することを特徴とする、請求項 1 から 4 のいずれかに記載のプログラム転送方法。

【請求項 6】 プログラム実行装置と、前記プログラム実行装置で実行されるプログラムを開発するための、前記プログラム実行装置と通信回線を介して結合可能なプログラム開発装置とを含むプログラム開発システムであって、前記プログラム開発装置は、前記プログラム実行装置で実行可能な形式のプログラムを開発するための手段と、前記実行可能な形式のプログラムおよびそのテストにおいて使用されるデータを記憶するための第 1 の記憶手段と、前記第 1 の記憶手段に格納された前記実行可能な形式のプログラムおよびデータを、前記通信回線を介して前記プログラム実行装置に逐次与え、前記実行可能な形式のプログラムを前記プログラム実行装置に実行させることにより、前記実行可能な形式のプログラムのデバッグを行なうためのデバッグ手段と、デバッグ済みの前記実行可能な形式のプログラムを前記通信回線を介して前記プログラム実行装置にロードするためのロード手段とを含み、前記プログラム実行装置は、

実行可能な形式のプログラムおよびプログラムの実行において使用されるデータを記憶するための第 2 の記憶手段と、

通信回線を介して前記プログラム開発装置に接続可能であり、テストモードが選択されたか否かに依存して、前記第 2 の記憶手段と、前記通信回線を介して接続される前記第 1 の記憶手段との一方を選択し、選択された記憶手段に記憶された実行可能な形式のプログラムを、前記選択された記憶手段を用いて実行するための実行手段と、

通信回線に接続可能であり、デバッグ済みのプログラムを前記プログラム開発装置の前記ロード手段から受信し、前記第 2 の記憶手段に格納するための格納手段とを含む、プログラム開発システム。

【請求項 7】 プログラム実行装置で実行されるプログラムを開発するための、通信回線を介してプログラム実行装置と結合可能なプログラム開発装置であって、プログラム実行装置で実行可能な形式のプログラムを開発するための手段と、

前記実行可能な形式のプログラムおよびそのテストにおいて使用されるデータを記憶するための記憶手段と、

前記記憶手段に格納された前記実行可能な形式のプログラムおよびデータを、前記通信回線を介して前記プログラム実行装置に逐次与え、前記実行可能な形式のプログラムを前記プログラム実行装置により実行させることにより、前記実行可能な形式のプログラムのデバッグを行なうためのデバッグ手段と、

デバッグ済みの実行可能な形式のプログラムを前記通信回線を介して前記プログラム実行装置にロードするためのロード手段とを含む、プログラム開発装置。

【請求項 8】 実行可能な形式のプログラムおよびプログラムの実行において使用されるデータを記憶するための記憶手段と、

通信回線を介してプログラム開発装置に接続可能であり、テストモードが選択されたか否かに依存して、前記記憶手段と、前記通信回線を介して接続されるプログラム開発装置に含まれる記憶装置との一方を選択し、選択された記憶手段または記憶装置に記憶された実行可能な形式のプログラムを実行するための実行手段と、

通信回線に接続可能であり、デバッグ済みのプログラムをプログラム開発装置から受信し、前記記憶手段に格納するための格納手段とを含む、プログラム実行装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 この発明は、通信回線を介してプログラム開発装置とターゲット装置とを接続してソフトウェアのデバッグを行なう際のプログラム転送に関し、特に、比較的低速な直列回線を介してプログラム開発装置とターゲット装置とを接続してソフトウェアのデバッグ作業を行なう際の作業の効率化に関する。

【0002】

【従来の技術】 図 12 を参照して、従来、プログラムを実行するターゲット装置 22 のプログラムを、表示装置 24、キーボード 26、およびハードディスクなどの外部記憶装置 28 が接続される開発装置 120 で開発し、通信回線 30 を介してターゲット装置 22 に与えることが行なわれている。

【0003】 開発装置 120 は、RAM (Random Access Memory) 46 と、通信回線 30 に接続された通信回路 48 と、表示装置 24、キーボード 26、外部記憶装置 28、および RAM 46 にそれぞれ結合されたコンパイラ 130、リンカ 132、およびデバッガ 134 とを含む。コンパイラ 130、リンカ 132、およびデバッガ 134 はソフトウェアの形式で設けられ、開発装置 120 に設けられた CPU により実行されてそれらの機能を実現する。

【0004】 ターゲット装置 22 は、CPU またはインタプリタ（以下常に CPU と呼ぶ。）60 と、CPU 60 および通信回線 30 に接続された通信回路 62 と、CPU 60 および通信回路 62 に接続された RAM 64 とを含む。

【0005】 図 12 に示す典型的な開発装置 120 とターゲット装置 22 では、ターゲット装置 22 で実行されるプログラムの開発は次のようにして行なわれている。まず、図 13 の最上段に示されるように、複数のソースプログラムファイル（図 13 に示す例ではファイル 1 ～ ファイル 5）を準備する。図 12 に示すコンパイラ 130 によってこれらソースファイルをコンパイルし、中間オブジェクトファイル 70a、72a、74a、76a、および 78a（図 13 中段）を生成する。さらにリンカ 132 によってこれら中間オブジェクトファイル 70a、72a、74a、76a、および 78a をリンクして、ターゲット装置 22 に転送する最終オブジェクトファイルを生成する（図 13 下段）。

【0006】 この場合図 13 に示すように、コンパイラで中間オブジェクトファイルを作成する場合、オブジェクトファイルのコード領域とデータ領域とはそれぞれ必要最低限のサイズとなるようにされている。また中間オブジェクトファイルをリンクする場合、先行するオブジェクトファイルが終了するとその直後に次のオブジェクトファイルをリンクするようにして最終的なオブジェクトファイルを生成している。したがってたとえば中間オブジェクトファイル 70a、72a、74a、76a、および 78a のサイズがそれぞれ 170 バイト、100 バイト、150 バイト、100 バイト、および 75 バイトであれば、最終ファイルのサイズはほぼこれらの和である 600 バイトとなる。

【0007】 従来このようにして作成した最終オブジェクトファイルをターゲット装置 22 に転送する場合、この最終のオブジェクトファイルの内容をそのまま転送す

ることが最もよく行なわれている。しかし、このように最終のオブジェクトファイルの内容をそのまま転送するのでは、転送データ量が多く転送時間がかかる。そのため、開発装置自体の処理速度が速くなったとしても転送時間により制約を受けるため、プログラムのデバッグに時間がかかり、その結果プログラム開発に要する時間も効率的に短縮化されないという問題点があった。

【0008】

【発明が解決しようとする課題】そのため、従来からプログラムの転送時間をより少なくする方式が求められていた。その1つの方式として、一旦プログラムをターゲット装置に転送した後、プログラムに修正が行なわれた場合、前回転送したオブジェクトファイルと、修正後のオブジェクトファイルとの差分を転送するものがある。これにより、変更のない部分は転送する必要がなく、データ転送量が減少する。

【0009】しかしこの場合、さらに解決すべき次のような問題がある。図14上段を参照して、元の最終的なオブジェクトファイルが中間オブジェクトファイル70a、72a、74a、76a、および78aであり、これらのサイズがそれぞれ175バイト、100バイト、150バイト、100バイト、および75バイトであったものとする。ファイル2のソースファイルを修正した後、これらをコンパイルした結果125バイトとなったものとする。オブジェクトをリンクすると、図14の下段に示すようにファイル2に対応するオブジェクトファイルの位置90a以降の領域140のすべてにおいて、ファイル2の修正前とは対応する部分のオブジェクトファイルの内容は変化している。したがって図14に示す例の場合には修正された部分以降の領域140のコードをすべて転送する必要がある。このような転送を頻繁に行なう必要がある場合、それほどデータ転送の時間が短縮されるとは言えない。

【0010】さらにまた、特開平5-197645号公報では、転送するデータ内に繰返しがある場合に、その繰返しを利用して転送データ量を削減する方式が提案されている。しかし処理対象となるデータの中にはこのような繰返しが発生する頻度は多いかもしれないが、プログラムコードの場合には内部に繰返しが存在することは比較的少ない。したがって特開平5-197645号公報の方式を用いてオブジェクトファイルを開発装置からターゲット装置に転送したとしても、転送効率が大幅に改善されるのは稀である。

【0011】オブジェクトファイルプログラムのターゲット装置への転送時間をできるだけ少なくすれば、ソフトウェアのデバッグ作業でのソースプログラムの修正、コンパイル、リンク、デバッグを通じたターンアラウンドタイムが大幅に改善される。したがって転送時間をできるだけ減少させることができるプログラム転送方式および装置が望まれている。

【0012】本発明はこのような問題を解決するためになされたもので、請求項1から5に記載の発明の目的は、転送時間を削減できるプログラム転送方法を提供することである。

【0013】また請求項6から8に記載の発明の目的はそれぞれ、プログラム開発の時間を削減できるプログラム開発システム、プログラム開発装置、およびプログラム実行装置を提供することである。

【0014】

【課題を解決するための手段】請求項1に記載の発明に係るプログラム転送方法は、相互にリンクされ、かつ各々がその内部に未使用領域を有する複数のオブジェクトファイルからなる実行可能形式のプログラムをプログラム開発装置からプログラム実行装置に転送し、プログラム実行装置の記憶装置上に展開する第1の転送ステップと、複数のオブジェクトファイルの少なくとも1つを変更して、変更前の大きさと予め定める関係を有する大きさの変更後オブジェクトファイルをプログラム開発装置において作成し、変更されたオブジェクトファイルを含む複数のオブジェクトファイルを、第1の転送ステップにおけるリンクの順番と予め定める関係にある順番に従ってリンクして変更後の実行可能形式のプログラムを作成するステップとを含む。リンク時の順番は、複数のオブジェクトファイルのうちの変更されないものが、変更後の実行可能形式のプログラム内において変更前と一致する位置となるように選択されている。このプログラム転送方法はさらに、変更後の実行可能形式プログラムのうち、変更されたオブジェクトファイルのみをプログラム実行装置に再び転送し、変更後のリンクの順番に従ってリンクするように記憶装置上に展開する第2の転送ステップを含む。

【0015】請求項2に記載の発明は、請求項1に記載のプログラム転送方法であって、作成するステップは、変更されるオブジェクトファイルに対応するソースプログラムをコンパイルして、オブジェクトプログラムを作成し、かつ変更後オブジェクトファイルの大きさが、予め定める単位長の倍数となるように、オブジェクトプログラムの末尾に所定長以上の大きさの未使用領域を付加して変更後オブジェクトファイルを生成するステップと、変更されたオブジェクトファイルを含む複数のオブジェクトファイルを、第1の転送ステップにおけるリンクの順番と予め定める関係にある順番に従ってリンクして変更後の実行可能形式のプログラムを作成するステップとを含む。リンク時の順番は、変更後オブジェクトファイルの大きさが変更前の大きさ以下であるときは第1の転送ステップにおいて転送される実行可能形式プログラムにおける順番と一致するように選択され、変更後オブジェクトファイルの大きさが変更前の大きさよりも大きいときには、変更後オブジェクトファイルが他のすべてのオブジェクトファイルの末尾に位置するように選択さ

れる。

【0016】請求項3に記載の発明は、請求項1記載のプログラム転送方法であって、作成するステップは、変更されるオブジェクトファイルに対応するソースプログラムをコンパイルして、オブジェクトプログラムを作成するステップと、変更されたオブジェクトプログラムを含む、複数のオブジェクトファイルに格納される複数のオブジェクトプログラムを、第1の転送ステップにおけるリンクの順番と予め定める関係にある順番に従って、かつ複数のオブジェクトプログラムのうちの変更されないものが、変更後の実行可能形式のプログラム内において変更前と一致する位置となるように、隣接するオブジェクトプログラム間に未使用領域が設けられるようにリンクして、変更後の実行可能形式のプログラムを作成するステップとを含む。リンク時の順番は、変更されたオブジェクトプログラムの大きさが変更前のオブジェクトファイルの大きさより小さいときは第1の転送ステップにおいて転送される実行可能形式プログラムにおける順番と一致するように選択され、変更されたオブジェクトプログラムの大きさが変更前のオブジェクトファイルの大きさよりも大きいときには変更後オブジェクトプログラムが他のすべてのオブジェクトファイルの末尾に位置するように選択される。

【0017】請求項4に記載の発明は、請求項1から3のいずれかに記載のプログラム転送方法であって、オブジェクトファイルに変更がなされたか否かを、当該オブジェクトファイルが前回転送されたときの日付と、当該オブジェクトファイルの作成日付とを比較することにより行なうことを特徴とする。

【0018】請求項5に記載の発明は、請求項1から4のいずれかに記載のプログラム転送方法であって、第2の転送ステップにおいて、変更されたオブジェクトファイルを変更前オブジェクトファイルと比較し、変更のあった部分のみをプログラム実行装置に転送することを特徴とする。

【0019】請求項6に記載の発明は、プログラム実行装置と、プログラム実行装置で実行されるプログラムを開発するための、プログラム実行装置と通信回線を介して結合可能なプログラム開発装置とを含むプログラム開発システムである。プログラム開発装置は、プログラム実行装置で実行可能な形式のプログラムを開発するための手段と、実行可能な形式のプログラムおよびそのテストにおいて使用されるデータを記憶するための第1の記憶手段と、第1の記憶手段に格納された実行可能な形式のプログラムおよびデータを、通信回線を介してプログラム実行装置に逐次与え、実行可能な形式のプログラムを、データに基づいてプログラム実行装置により実行させることにより、実行可能な形式のプログラムのデバッグを行なうためのデバッグ手段と、デバッグ済みの実行可能な形式のプログラムを通信回線を介してプログラム

実行装置にロードするためのロード手段とを含む。プログラム実行装置は、実行可能な形式のプログラムおよびプログラムの実行において使用されるデータを記憶するための第2の記憶手段と、通信回線を介してプログラム開発装置に接続可能であり、テストモードが選択されたか否かに依存して、第2の記憶手段と、通信回線を介して接続される第1の記憶手段との一方を選択し、選択された記憶手段に記憶された実行可能な形式のプログラムを、選択された記憶手段を用いて実行するための実行手段と、通信回線に接続可能であり、デバッグ済みのプログラムをプログラム開発装置のロード手段から受信し、第2の記憶手段に格納するための格納手段とを含む。

【0020】請求項7に記載の発明に係るプログラム開発装置は、プログラム実行装置で実行可能な形式のプログラムを開発するための手段と、実行可能な形式のプログラムおよびそのテストにおいて使用されるデータを記憶するための記憶手段と、記憶手段に格納された実行可能な形式のプログラムおよびデータを、通信回線を介してプログラム実行装置に逐次与え、実行可能な形式のプログラムを、データに基づいてプログラム実行装置により実行させることにより、実行可能な形式のプログラムのデバッグを行なうためのデバッグ手段と、デバッグ済みの実行可能な形式のプログラムを通信回線を介してプログラム実行装置にロードするためのロード手段とを含む。

【0021】請求項8に記載の発明は、実行可能な形式のプログラムおよびプログラムの実行において使用されるデータを記憶するための記憶手段と、通信回線を介してプログラム開発装置に接続可能であり、テストモードが選択されたか否かに依存して、記憶手段と、通信回線を介して接続されるプログラム開発装置に含まれる記憶装置との一方を選択し、選択された記憶手段または記憶装置に記憶された実行可能な形式のプログラムを実行するための実行手段と、通信回線に接続可能であり、デバッグ済みのプログラムをプログラム開発装置から受信し、記憶手段に格納するための格納手段とを含む。

【0022】

【作用】請求項1に記載の発明に係るプログラム転送方法によれば、オブジェクトファイルは、変更されないオブジェクトファイルが、実行可能な形式のプログラム内において変更前と一致する位置となるような順番でリンクされる。変更されたオブジェクトファイルのみをプログラム実行装置に再び転送し、転送済みの他のオブジェクトファイルとある順番に従ってリンクするだけで、変更後の実行可能な形式のプログラムをプログラム実行装置内に生成できる。各オブジェクトファイル内に予め未使用領域を設けているので、オブジェクトプログラムサイズが大きくなっても、未使用領域の大きさを調節すれば、オブジェクトファイルの大きさを変更前と一致させることができ、各オブジェクトファイルの位置を変更せ

ずに上記したリンクを行なうことが可能である。

【0023】請求項2に記載の方法では、変更後のオブジェクトプログラムの末尾に所定長以上の大きさの未使用領域を付加して、オブジェクトファイルの大きさが、予め定める単位長の倍数となるようにする。オブジェクトファイルをリンクするときの順番は、変更後オブジェクトファイルの大きさが変更前の大きさ以下であるときは前回転送時の順番と一致するので、変更前のオブジェクトファイル部分を変更後のオブジェクトファイルで入換えるだけで、変更後の実行可能プログラムを生成できる。変更後オブジェクトファイルの大きさが変更前の大きさよりも大きいときには、変更後オブジェクトファイルが他のオブジェクトファイルの末尾に配置されるので、他のオブジェクトファイルの位置を変化させる必要はない。

【0024】請求項3に記載の方法によれば、変更されないオブジェクトプログラムの実行可能形式のプログラム内の位置が変更前と一致するように、かつ隣接するオブジェクトプログラム間の未使用領域が設けられるようにオブジェクトプログラムをリンクして、実行可能形式のプログラムが作成される。変更されたオブジェクトプログラムが変更前のオブジェクトファイルより小さいときは第1の転送ステップにおいて転送される実行可能形式プログラムにおける順番と一致する順番でオブジェクトファイルがリンクされるので、変更前のオブジェクトファイル部分を変更後のオブジェクトファイルで入換えるだけで、変更後の実行可能プログラムを生成できる。変更されたオブジェクトプログラムが変更前のオブジェクトファイルの大きさよりも大きいときには、変更後オブジェクトプログラムが他のオブジェクトファイルの末尾に位置するようにオブジェクトファイルがリンクされるので、他のオブジェクトファイルの位置は変化させる必要はない。

【0025】請求項4に記載の発明では、オブジェクトファイルが前回転送されたときの日付と、当該オブジェクトファイルの作成日付とを比較し、オブジェクトファイルの作成日付が新しいもののみが転送されるので、転送量が少なくなるとともに、最新のプログラムをプログラム実行装置に転送できる。

【0026】請求項5に記載の方法によれば、変更されたオブジェクトファイルのうち、変更のあった部分のみが転送されるので、変更されたオブジェクトファイル全体を転送する場合よりさらに転送量が削減できる。

【0027】請求項6に記載のシステムによれば、テストモードが選択されると、プログラム開発装置の第1の記憶手段に格納された実行可能な形式のプログラムおよびデータが、通信回線を介してプログラム実行装置に与えられ、プログラム実行装置により実行される。実行結果はプログラム開発装置のデバッグ手段に与えられ、プログラムのデバッグが行なわれる。開発の完了していな

いプログラムをプログラム実行装置にロードしなくてもそのテストが行なえる。

【0028】請求項7に記載の発明に係るプログラム開発装置によれば、プログラム実行装置で実行可能な形式のプログラムを開発し、データとともに記憶手段に記憶させれば、プログラムをプログラム実行装置にロードしなくても、通信回線を介してプログラム実行装置に逐次与えて実行させることができる。プログラムをプログラム開発装置のデバッグ手段によりデバッグし、デバッグが終了した時点でプログラム実行装置にロードすればよく、開発時のプログラムのロードにかかる時間を削減できる。

【0029】請求項8に記載の発明に係るプログラム実行装置によれば、プログラムをプログラム実行装置にロードしなくても、通信回線を介してプログラム実行装置に逐次与えて実行させることができる。プログラムのデバッグは通信回線を介して外部のプログラム開発装置に結果を与えて行なえばよく、テストの都度、実行可能な形式のプログラムをプログラム実行装置にロードする必要がない。そのためテストのための時間を削減できる。

【0030】

【実施例】図1に、本発明に係るプログラム開発システムの構成を示す。この構成は図12に示す従来のシステムと概略同じであるが、図12のコンパイラ130、リンカ132、デバッガ134に代えてそれぞれ後述するような機能を有するコンパイラ40、リンカ42およびデバッガ44を含むことを特徴とする。

【0031】コンパイラ40、リンカ42およびデバッガ44の機能について以下にこのシステムの動作とともに説明する。まず、開発装置20からターゲット装置22にプログラム・データを転送する場合の例を説明する。ここでは、例として5つのソースファイル（ファイル1、ファイル2、ファイル3、ファイル4、およびファイル5）があり（図2上段）、それぞれをコンパイルした後相互にリンクしてターゲット装置22で動作するオブジェクトプログラムを作成する場合について説明する。

【0032】図2を参照して、本実施例のコンパイラ40は、ソースファイルをコンパイルした後、その末尾に未使用領域を付加し、生成される中間オブジェクトファイルの大きさが所定の単位長の倍数となるようにする。本実施例の場合には、作成される中間オブジェクトファイルの大きさが50バイトの倍数となるように各オブジェクトプログラムの末尾に未使用領域を付加する。たとえばファイル1～5をコンパイルした結果の中間オブジェクトプログラムがそれぞれ中間ファイル70a、72a、74a、76a、および78aであり、その長さが175バイト、100バイト、150バイト、100バイト、および75バイトであるものとする（図2中段）。この場合コンパイラ40は、各領域の末尾に未使



用領域 70 b、72 b、74 b、76 b、および 78 b を付加して、それぞれ 200 バイト、150 バイト、200 バイト、150 バイト、および 100 バイトとなるようにする。こうして中間オブジェクトファイル 70、72、74、76、および 78 が得られる。

【0033】さらにリンク 132 は、これら中間オブジェクトファイル 70、72、74、76、および 78 を従来と同様にこの順番に従ってリンクする。したがって図 2 下段に示されるように、最終的なオブジェクトファイルの大きさは 800 バイトとなる。

【0034】コンパイラの機能をこのように変更することで、次に述べるようにプログラムのデバッグ時に転送するプログラム量を削減できる。図 3 を参照して、たとえばファイル 2 の内容を変更した場合を考える。そしてこれをコンパイルした結果、修正前のオブジェクトプログラムサイズ 100 バイト (72 a) が 125 バイト (82 a) に増加したものとする (図 3、3 段目)。この場合にも、前述のとおりファイル 2 の中間オブジェクトファイルのサイズは変更前で 150 バイトあり、そのうちの末尾の 50 バイトが未使用領域であったのだから、最後の 50 バイトのうちの最初の 25 バイトを、ファイル 2 に対応するオブジェクトプログラムの増加分として使用する。こうしても、ファイル 2 に対応するオブジェクトファイル全体の大きさは 150 バイトで変わらない。したがってこれを従来と同様にリンクすることにより、図 3 の最下段に示されるオブジェクトファイルを得ることができる。

【0035】このオブジェクトファイルのうち、領域 82 a 以外の部分は図 3 の 4 段目に示される修正前の最終オブジェクトファイルと全く同様である。ターゲット装置には、前回転送した内容が残っている。したがってこのファイル 2 に対応するオブジェクトファイル部分である 150 バイトのみをターゲット装置に転送し、転送された部分をその他の領域とリンクすれば、修正後の最新オブジェクトファイルをターゲット装置に生成することができる。この場合のリンクの順番は最初に転送したときと同じである。

【0036】このように、変更後の中間オブジェクトプログラムのサイズが、前回設定したオブジェクトファイル (オブジェクトプログラムと未使用領域との和) の大きさ以内に収まるのであれば、ファイル 3 以降の位置を変化させる必要はない。そして、変更のあった部分のみを転送することによりターゲット装置に変更後のプログラムを生成することができる。なお、どのオブジェクトファイルを転送するかは、前回各オブジェクトファイルを転送した日付および時刻と、今回処理対象となるオブジェクトファイルの作成日付、時刻とを比較し、新しくなっているもののみを選択することで決めればよい。

【0037】図 3 および図 14 の従来例を比較すると、従来では 200 バイトの以降 625 バイトまでの 4

00 バイトを転送する必要があった。それに比較して図 3 に示される実施例の場合、転送量は 150 バイトのみであり、転送量が大幅に削減される。

【0038】修正を行なった結果、前回生成されたオブジェクトファイルの大きさよりも、中間オブジェクトプログラムのサイズが大きくなった場合 (図 4 (A)) を考える。この場合、修正前のオブジェクトプログラムと付加された未使用領域とを合計した領域の大きさよりも、修正後のオブジェクトプログラムの方が大きい。したがって、従来と同様の順番でリンクしたのでは、図 4 (B) の下段に示すように、ファイル 2 に対応するオブジェクトファイル領域 90 a 以降の部分がすべてずれ、その結果ファイル 2 以降のすべてを転送する必要が生じる。これでは従来と同様の結果となってしまうので、これを改善することが望ましい。

【0039】プログラムのうち変更された部分はファイル 2 の内容だけである。したがって本来ならばファイル 2 に対応するオブジェクトファイルの内容だけをターゲット装置に転送すればよいはずである。にもかかわらず図 4 (B) の下段に示されるような状況が発生してしまうのは、最初のリンク時と同じ順番に従ってリンク手続を行なうためである。そこで本発明では、それぞれのファイルの内容を前回のリンク時にはどのアドレスに配置したかを記録しておき、修正のあった部分が、修正の結果次の領域にまでくい込んでしまう場合には、前回のリンク時の順番には従わず、修正のあったものを全オブジェクトファイルの最後にリンクする方式をとる。

【0040】このようにしてリンクした結果を図 5 に示す。図 5 (A) 下段に示すように、ファイル 2 を修正した結果 175 バイトのオブジェクトプログラム 90 a が生成されたものとする。この場合、ファイル 2 に対応するオブジェクトファイルの大きさは 200 バイトとなる。その結果、図 5 (B) の上段に示される、修正前のオブジェクトファイルの領域 72 a、72 b の合計長 (150 バイト) を超えてしまう。この場合にはこの領域 90 a を図 5 (B) の下段に示されるように他のすべてのオブジェクトファイルの最後の位置に移す。他のオブジェクトファイルのリンク位置は変更しない。こうすることにより、図 5 (B) の下段の最終部分に示される 175 バイトの領域 9.2 のデータのみをターゲット装置に転送すればよい。したがってこの場合にもデータの転送量は従来と比べて著しく削減できる。

【0041】なお以上の第 1 の実施例では、ソースファイルをコンパイルする場合に、オブジェクトプログラムの末尾に、コンパイラが未使用領域を付加するようにした。しかし本発明はこれには限定されず、リンクを用いて同じ機能を実現させることもできる。図 7 を参照して、ソースファイル 1 ~ 5 から、従来と同様のコンパイラにより中間ファイル 70 a、72 a、74 a、76 a、および 78 a を得るものとする。この各中間オブ



エクトファイルの大きさはそれぞれ図13に示される例と同じく175バイト、100バイト、150バイト、100バイト、および75バイトであるものとする。

【0042】図7の最下段に示すように、これら中間オブジェクトファイルをリンクする際に、各オブジェクトファイルの間に未使用領域70b、72b、74b、76b、および78bを挿入されるようにリンクを行なうようにする。こうすることによっても、コンパイラで未使用領域を付加する場合と同じ結果が得られる。もちろん、修正の結果得られるオブジェクトファイルの大きさが、修正前のオブジェクトプログラムのサイズと、挿入された未使用領域のサイズとの和よりも大きい場合には、そのオブジェクトファイルは最終オブジェクトファイルの末尾に付加するものとする。

【0043】図6に、このときのリンクの手の流れをPDSチャート形式で示す。図6を参照して、まずすべてのリンク対象ファイルについて以下の処理を行なう。まずリンク対象ファイルのサイズを調べる。調べた結果と前回のリンク結果と比較し、次の領域に~~く~~い込んでしまいか否かを判断する。次の領域までリンク参照ファイルが~~く~~い込む場合にはそのファイルをリンクせずに未処理テーブルに記憶する。そしてリンクアドレスを前回のアドレスまで進める。この処理はリンク対象ファイルの直前のリンク対象ファイルの未使用領域を、そのリンク対象ファイルの領域分だけ増加させることを意味する。

【0044】次の領域に~~く~~い込まないと判断された場合にはそのファイルを現在のアドレスを先頭としてリンクし、このファイルのサイズに、全体のファイル長が前述したように50バイトの倍数となるように未使用領域を付加しリンクアドレスとする。こうして決まったリンクアドレス（リンク対象ファイルの最終アドレス）をしきい値と呼ぶことにする。

【0045】以上の処理をすべてのリンク対象ファイルについて繰返す。この処理が終了すると、今度はすべての未処理テーブルについて次の処理を行なう。

【0046】まずそのファイルを従来と同様にリンクする。そしてリンクアドレスにそのリンク対象ファイルのサイズを加算し、さらに前述したようにそのファイルサイズの全体が50バイトの倍数となるように未使用領域を付加しそれを新たなリンクアドレスとする。この処理をすべての未処理テーブルのファイルについて繰返し行なう。この処理が終了するとリンク処理は完了である。

【0047】以上のようにすることにより、変更のあったオブジェクトファイル領域のみをターゲット機器に転送すればターゲット機器側で変更後のオブジェクトファイルを生成することができる。プログラムの転送量を削減することができ、その結果転送時間も減少しプログラムの開発を効率化できる。

【0048】なお、以上の実施例では、ソースファイルにより規定される各オブジェクトファイルを1単位とし

て転送するか否かを定める方法を説明してきた。しかし本発明はこれには限定されず、修正があった1つの中間オブジェクトファイルの内部でも、変更がされた部分と変更のない部分とを調べ、変更がない部分は転送しないようにすることも可能である。つまり、1単位オブジェクトファイルの修正前と修正後とを比較し、変更のあった部分のみを転送し、変更のなかった部分は転送しないものとする。これによりさらにデータの転送量が削減できる。

【0049】こうした動作を実現する手順を図8に示す。図8を参照して、まず全体処理の開始に先立って修正フラグをリセットしておく。修正フラグは、現在処理中の位置が前回と異なっているか否かを示すためのものである。

【0050】続いてロード対象の内容のすべてについて以下の処理を繰返し行なう。まず、プログラム中、ロードする位置の内容が前回と異なっているか否かを判定する。異なっている場合にはその位置を記録し、修正フラグをセットする。前回と異なっていない場合には、修正フラグがセットされているか否かを調べる。セットされていれば、前回覚えていた位置から現在のデータまでをターゲット装置に対して転送し、修正フラグをリセットする。修正フラグがセットされていない場合には次のロード対象の内容についての処理に移る。

【0051】以上の処理をすべてのロード対象の内容について行なうことにより、変更があった部分のみをターゲット装置に転送できる。

【0052】デバッグ時には大きな修正が少なく、小さな修正が頻繁に行なわれる。またプログラムのローディングには、低速回線しか用いられないことが多い。こうした事実を鑑みると、デバッグのプログラムの修正による最終オブジェクトプログラムの変更は局所的なものであり、したがって上述したような方法を用いてターゲット装置にプログラムの変更分のみを転送するようにすれば、最終オブジェクトプログラムのターゲット機器へのローディング時間を大幅に短縮できる。

【0053】一方、このようなプログラムの転送と呼ばれる処理自体をなくす方法も考えられる。以下そうした実施例を説明する。この実施例の説明に先立ち、従来のターゲット装置でどのようにして命令を実行解釈しているかについて図9を参照して説明する。図9を参照して、PC（プログラムカウンタ）により指定されるアドレスに従って、次の命令をRAMまたはROMから取出し、各命令ごとの処理を行ない、こうした処理を繰返し行なっていく。

【0054】こうして処理される命令には様々なものがあるが、大別すると次の2つの種類に分かれる。

【0055】（1） 解釈実行手段（CPU）の内部での演算処理

（2） RAM上のデータに対する読込または書込動作

そこで本発明では、解釈実行手段が、命令やデータを格納したメモリにアクセスする場合、特定のテストモードが選択された場合に、ターゲット装置のメモリではなく、通信回路および通信回線を介して開発装置にその値を取出して、値を得る手段を追加する。この構成を図 10 に示す。図 10 のシステムは、図 1 のシステムと類似しているが、図 1 の CPU またはインタプリタ 60 に代えて、テストモードが選択された場合に RAM 46 ではなく通信回路 114 および通信回線 30 を介して開発装置から命令およびデータを読出すように動作するインタフェース 116 を含んでいる。また図 10 の開発装置 100 のデバッガ 44 は、通信回路 110 を介してターゲット装置 102 から与えられる命令またはデータの読出／書込命令に回答して、RAM 46 からのデータの読出および RAM 46 へのデータの書込を行なう。

【0056】このシステムではプログラムのデバッグが次の手順で行なわれる。

(1) 従来の手法と同じ手法で、ターゲット装置 102 に転送する内容を開発装置 100 で作成する。

【0057】(2) 開発装置 100 のデバッガ 44 が、転送するデータを読込む。

(3) 開発装置 100 から、ターゲット装置 102 の CPU またはインタプリタ 112 を起動する。この場合テストモードが選択され、インタフェース 116 は通信回路 114 を介して、命令やデータの読出／書込を開発装置 100 に対して問合せることになる。

【0058】(4) ターゲット装置 102 の CPU またはインタプリタ 112 が、次の命令を開発装置 100 に問合せる。

【0059】(5) 開発装置 100 は、問合せたアドレスに対応する命令を RAM 46 から読出し、ターゲット装置 102 に返す。

【0060】(6) ターゲット装置 102 は、戻ってきた命令に対する処理を呼出し実行する。

【0061】(7) もしもメモリ上へのアクセスであれば、データはインタフェース 116 により開発装置 100 に問合せられる。

【0062】(8) 開発装置 100 は、ターゲット装置 102 から戻ってきた内容をデータとして取扱い、必要であれば RAM 46 に保存する。

【0063】このようにして、ターゲット装置に最終オブジェクトプログラムをその都度ロードすることなく、プログラムのデバッグ作業が行なえる。データ転送に必要な時間が不要となるため、デバッグ効率を高めることができる。

【0064】なお、図 10 の CPU またはインタプリタ 112 およびインタフェース 116 による動作の手順を図 11 に示す。図 11 の処理の内容は、次の命令を通信手段を用いて開発装置 100 から取出すことを除き、図 9 に示した処理の内容と同様である。

【0065】

【効果】以上のように請求項 1 に記載の発明によれば、変更されたオブジェクトファイルのみをターゲット機器に再び転送し、転送済みの他のオブジェクトファイルとある順番に従ってリンクするだけで、変更後の実行可能な形式のプログラムをプログラム実行装置内に生成できる。変更後の実行可能な形式のプログラムの全体を転送する必要がない。その結果、転送時間を削減できるプログラム転送方法を提供できる。

【0066】請求項 2 に記載の発明によれば、転送後オブジェクトファイルの大きさが変更前の大きさ以下であるときは前回転送時の順番と同じ順番でオブジェクトファイルをリンクするので、変更前のオブジェクトファイル部分を変更後のオブジェクトファイルで入換るだけで、変更後の実行可能プログラムを生成できる。変更後オブジェクトファイルの大きさが変更前の大きさよりも大きいときには、変更後オブジェクトファイルが他のオブジェクトファイル末尾に配置されるので、他のオブジェクトファイルの位置を変化させる必要はない。したがって、他の変更前オブジェクトファイルは転送せずに、変更後の実行可能な形式のプログラムをプログラム実行装置内に生成できる。その結果、転送時間を削減できるプログラム転送方法を提供できる。

【0067】請求項 3 に記載の方法によれば、変更されたオブジェクトプログラムが変更前のオブジェクトファイルより小さいときは第 1 の転送ステップにおいて転送される実行可能形式プログラムにおける順番と一致する順番でオブジェクトファイルがリンクされるので、変更前のオブジェクトファイル部分を変更後のオブジェクトファイルで入換るだけで、変更後の実行可能プログラムを生成できる。変更されたオブジェクトプログラムが変更前のオブジェクトファイルの大きさよりも大きいときには、変更後オブジェクトプログラムが他のオブジェクトファイルの末尾に位置するようにオブジェクトファイルがリンクされるので、他のオブジェクトファイルの位置を変化させる必要はない。したがって、他の変更のないオブジェクトファイルは転送せずに、変更後の実行可能な形式のプログラムをプログラム実行装置内に生成できる。その結果、転送時間を削減できるプログラム転送方法を提供できる。

【0068】請求項 4 に記載の発明では、オブジェクトファイルの作成日付の新しいもののみが転送されるので、転送量が少なくなるとともに、最新のプログラムをプログラム実行装置に生成できる。その結果、最新のプログラムを削減された転送時間で転送できるプログラム転送方法を提供できる。

【0069】請求項 5 に記載の方法によれば、変更されたオブジェクトファイルのうち、変更のあった部分のみが転送されるので、さらに転送量が削減できる。その結果、転送時間をさらに削減できるプログラム転送方法を

提供できる。

【0070】請求項6に記載の発明によれば、開発の完了していないプログラムをプログラム実行装置にロードしなくてもそのテストが行なえる。したがって、プログラム開発時のプログラムの転送時間を削減できる。その結果、プログラム開発の時間を削減できるプログラム開発システムを提供できる。

【0071】請求項7に記載の発明に係るプログラム開発装置によれば、テストの都度、実行可能な形式のプログラムをプログラム実行装置にロードする必要がなく、開発時のプログラムのロードにかかる時間が削減できる。その結果、プログラム開発の時間を削減できるプログラム開発装置を提供できる。

【0072】請求項8に記載の発明に係るプログラム実行装置によれば、テストの都度、実行可能な形式のプログラムをプログラム実行装置にロードする必要がなく、テストのための時間を削減できる。その結果、プログラム開発の時間を削減できるプログラム開発装置を提供できる。

【図面の簡単な説明】

【図1】本発明の一実施例のプログラム開発システムのブロック図である。

【図2】本発明の一実施例によるプログラムのコンパイルおよびリンクの形式を模式的に示す図である。

【図3】本発明の一実施例の効果を従来例と比較して説明する模式図である。

【図4】変更後オブジェクトプログラムのサイズが大きくなった場合の状態を模式的に示す図である。

【図5】変更後のオブジェクトプログラムサイズが大きくなった場合の、本発明によるリンク方法を示す模式図である。

【図6】本発明に係るリンク方法の手順を説明する図である。

【図7】本発明の一実施例において、リンクにより所定の形式の最終オブジェクトファイルを作成する方法を模式的に示す図である。

【図8】修正部分のみを転送する場合の手順を示す図である。

【図9】通常動作時のターゲット装置の動作手順を示す図である。

【図10】本発明の第2の実施例のプログラム開発システムのブロック図である。

【図11】本発明の第2の実施例による処理手順を模式的に示す図である。

【図12】従来のプログラム開発システムのブロック図である。

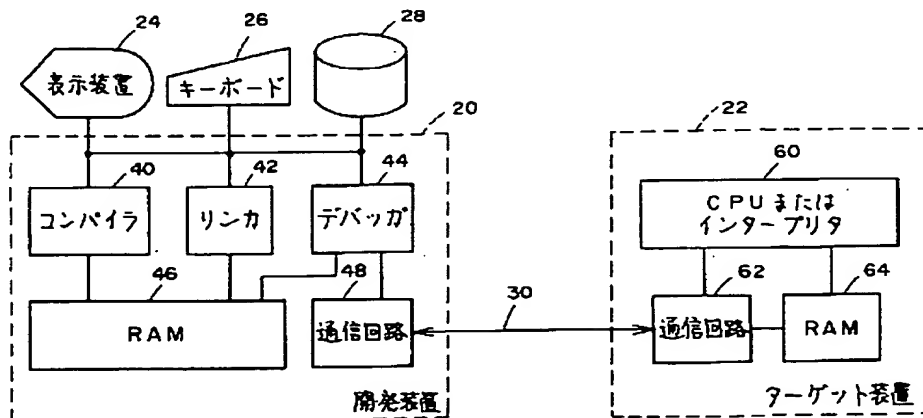
【図13】従来のプログラム開発システムによる最終オブジェクトファイルの作成方法を模式的に示す図である。

【図14】従来のシステムにおける、プログラムの一部修正の場合の最終オブジェクトファイルの変化を模式的に示す図である。

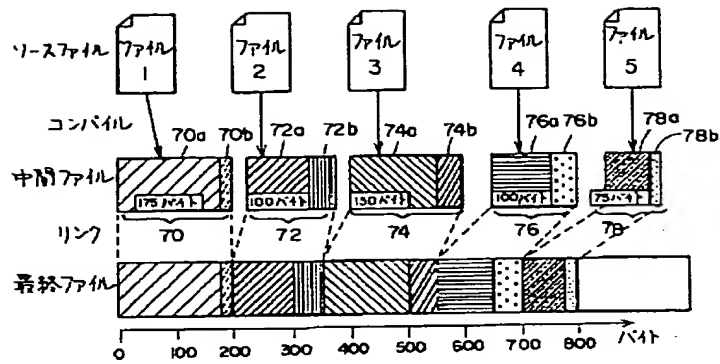
【符号の説明】

20、100 開発装置  
22、102 ターゲット装置  
40 コンパイラ  
42 リンカ  
44 デバッガ  
46 RAM  
48、62、110、114 通信回路  
60、112 CPU  
64 RAM

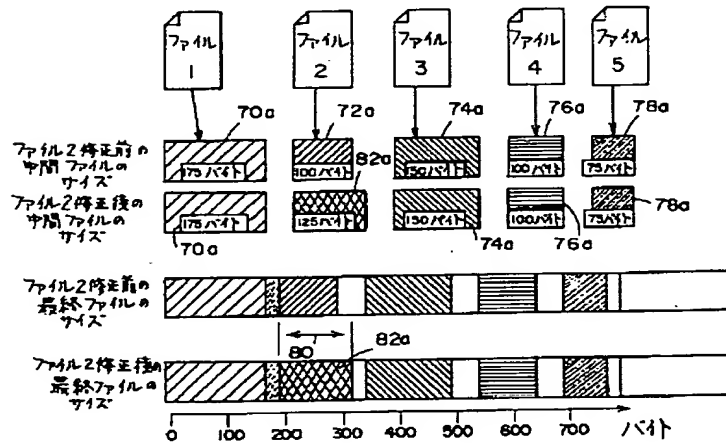
【図1】



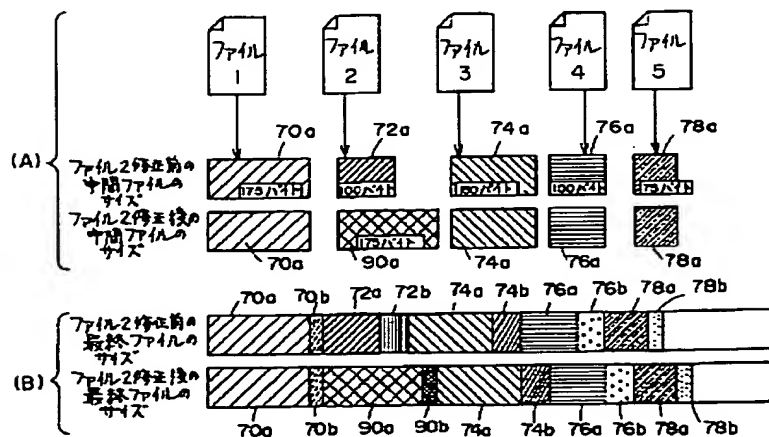
【図 2】



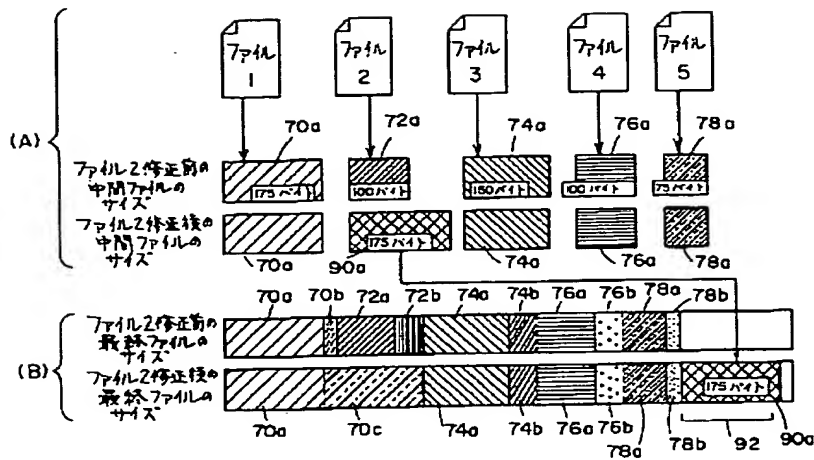
【図 3】



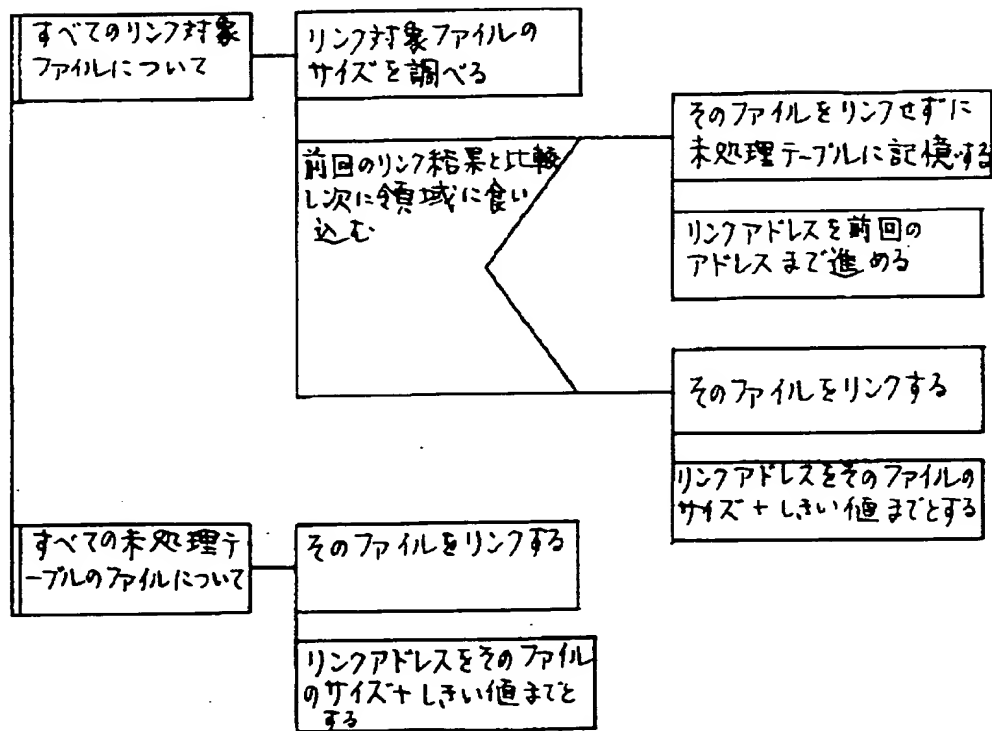
【図 4】



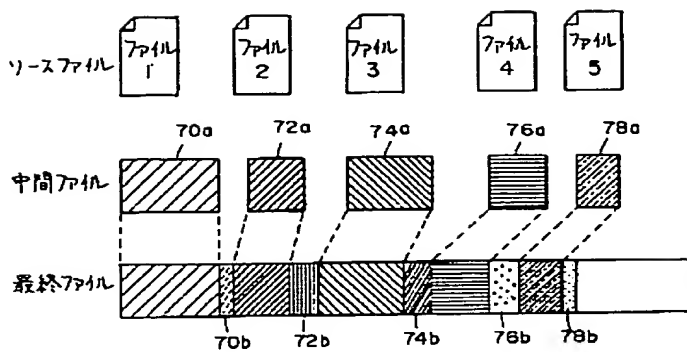
【図 5】



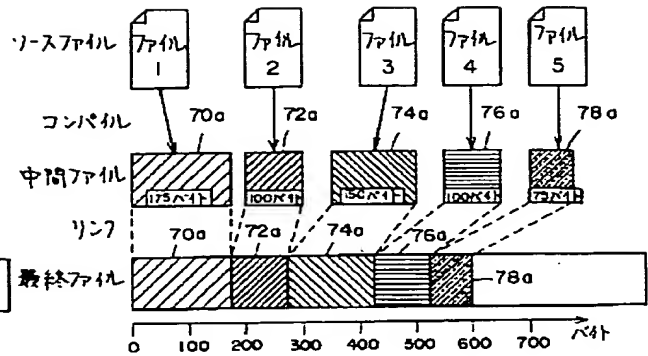
【図 6】



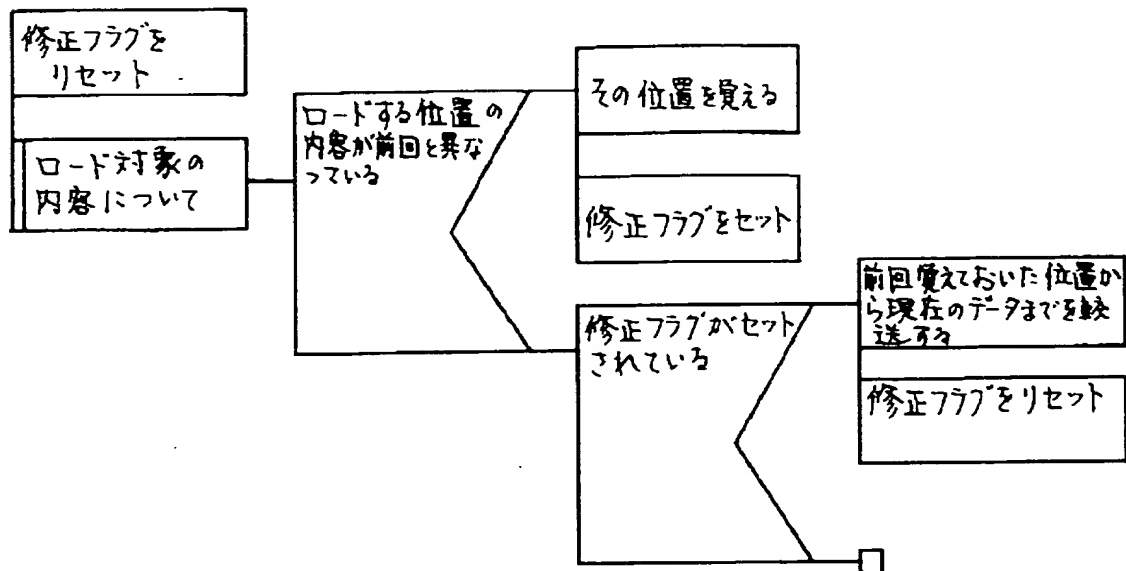
【図 7】



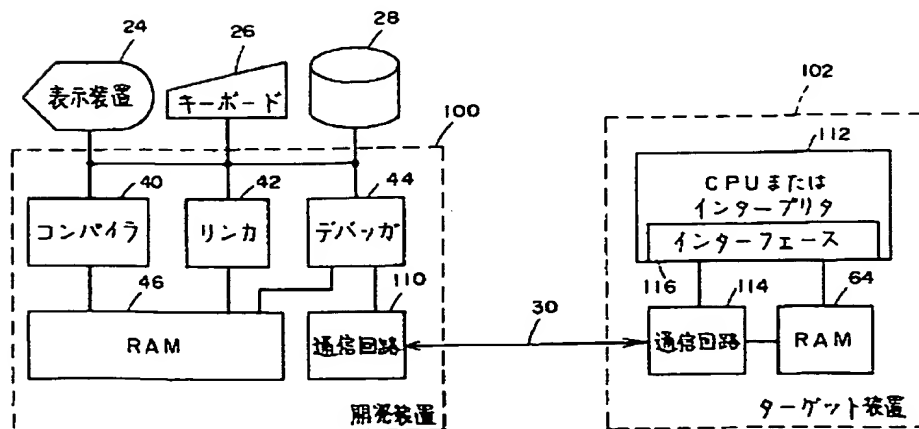
【図 13】



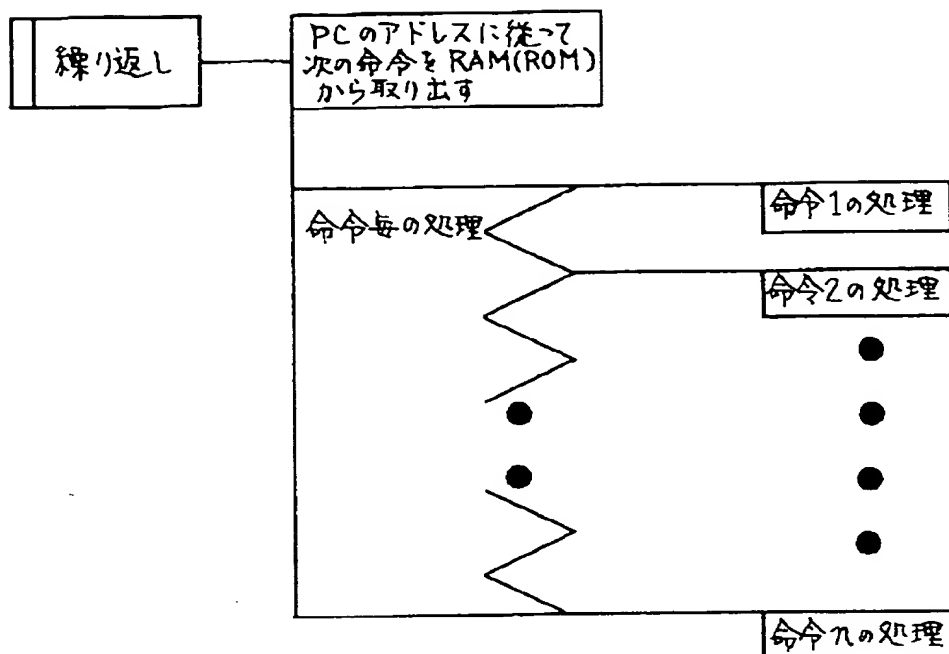
【図 8】



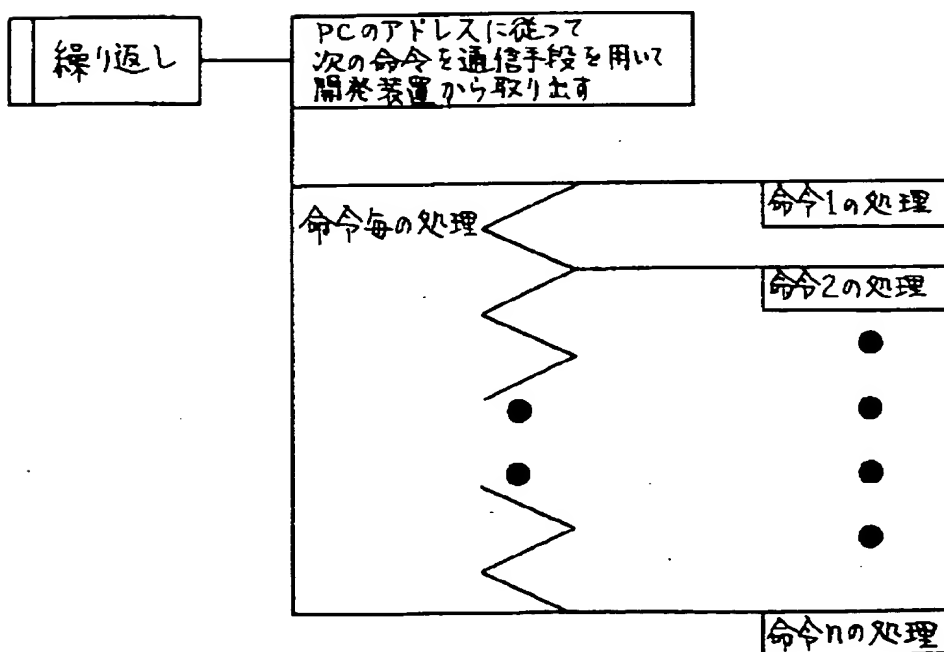
【図 10】



【図 9】

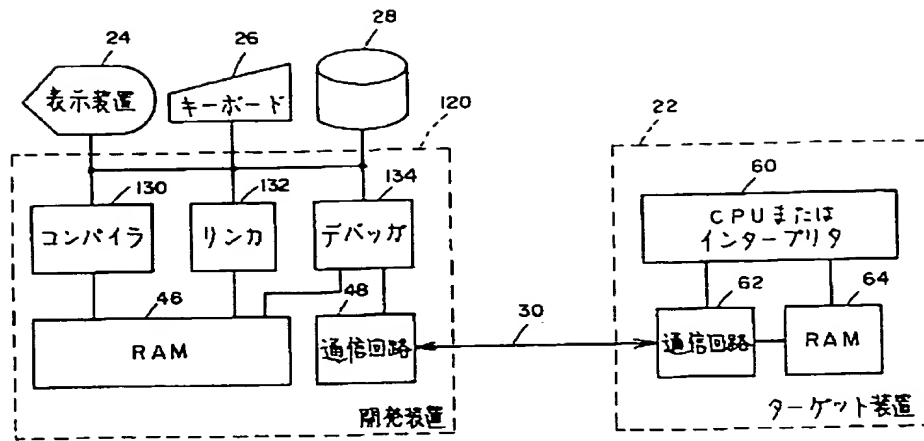


【図 11】





【図12】



【図14】

